

Advancing Predictive Control: Insights from Maze Exploration Using Markov Decision Processes

Robel Asgedom¹, Igor Korobiichuk²

¹ Ukraine Independent Researcher, Łódź, Poland,

² Warsaw University of Technology, plac Politechniki 1, 00-661, Warsaw, Poland

Abstract

Predictive control plays a significant role in mobile robotics, especially in trajectory tracking, obstacle avoidance, and real-time decision-making. In this study, we explore how Markov Decision Processes (MDPs) can be integrated with predictive control to enhance navigation, particularly in maze-like environments. A case study on MDP-based maze exploration analyzes key system limitations, including computational complexity and real-time adaptability. While MDPs often struggle to adapt to dynamic environments, predictive techniques like Model Predictive Control (MPC) offer improvements in trajectory optimization and responsiveness. We also discuss practical applications in areas such as warehouse navigation and multi-robot coordination, showing the benefits of combining MDPs and predictive control for robust performance in real-world scenarios.

Keywords

Predictive Control, Markov Decision Processes, Maze Exploration, Mobile Robots, Trajectory Tracking

1. Introduction

Autonomous navigation is a fundamental capability in mobile robotics, allowing robots to traverse complex and dynamic environments efficiently. Achieving accurate trajectory tracking and efficient maze exploration is still challenging due to uncertainties in the environment, sensor limitations, and computational constraints. Addressing these challenges requires robust decision-making frameworks and control techniques.

Predictive control techniques, particularly Model Predictive Control (MPC), have demonstrated significant advantages in trajectory tracking and obstacle avoidance by enabling real-time adjustments based on predicted future states [1,2]. Its structured approach has seen success in autonomous driving, industrial automation, and robotic path planning, offering a structured approach to real-time motion optimization while ensuring the satisfaction of the constraints. In parallel, Markov Decision Processes (MDPs) offer a robust mathematical foundation for decision-making under uncertainty, widely applied in navigation and mapping tasks [3,4].

Successes of MDPs and MPC are well documented, but their integration in mobile robotics is still underexplored. Existing studies primarily focus on standalone MDPs for decision-making or MPC for trajectory optimization, yet few works have attempted to bridge the gap between these two methods. Most of the literature on MDPs addresses static environments with predefined state transitions, limiting their real-time adaptability. Although MPC offers dynamic control it lacks the high-level policy optimization capabilities of MDPs. To overcome the limitations, the article examines integrating MDPs with predictive control techniques. We aim to combine MDP-based decision-making with the real-time adaptability of MPC to enhance mobile robot trajectory tracking in dynamic and uncertain environments.

This study extends previous work by analyzing the limitations of MDP-based maze exploration and demonstrating how predictive control can address these challenges. We highlight the novelty of our approach by reviewing existing literature and identifying gaps in current research. Researchers have extensively studied individual applications of MDPs and MPC, yet their combined use to enhance real-time adaptability and decision-making in maze exploration remains underexplored.

¹CMIS-2025: Eighth International Workshop on Computer Modeling and Intelligent Systems, May 5, 2025, Zaporizhzhia, Ukraine

✉ robelasgedom629@gmail.com (R. Asgedom); igor.korobiichuk@pw.edu.pl (I. Korobiichuk)



0009-0007-2330-3345 (R. Asgedom); 0000-0002-5865-7668 (I. Korobiichuk)



© 2025 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Primarily, this article aims to contribute to this area by presenting a structured approach for integrating predictive control with MDP-based systems.

The rest of this paper is structured as follows. Section II reviews related work, analyzing existing MDP and predictive control approaches in mobile robotics. Section III presents the case study, discussing the implementation of MDPs for maze exploration. Section IV explores the integration of predictive control techniques and their impact on real-time navigation. Finally, Section V outlines future research directions and potential improvements in hybrid MDP-MPC frameworks.

2. Background and Related Work

2.1. Markov Decision Processes in Robotics

Markov Decision Processes (MDPs) provide a mathematical framework to model decision-making problems in stochastic environments [1]. An MDP is defined as a tuple $(\mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{R}, \gamma)$, where:

- \mathbf{S} is a finite set of states representing the possible configurations of the environment.
- \mathbf{A} is a finite set of actions available to the agent.
- $\mathbf{P}(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ is the state transition probability, which defines the probability of reaching the state.
- $\mathbf{R}(\mathbf{s}, \mathbf{a})$ is the reward function, which assigns a scalar reward to each state-action pair.
- $\gamma \in [0,1]$ is the discount factor, which determines the importance of future rewards.

The objective in an MDP is to find an optimal **policy** $\pi(\mathbf{s})$, which maps state to actions to maximize the expected cumulative reward:

$$V^\pi(s) = E \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right],$$

where $V^\pi(s)$ is the value function representing the expected reward when following **policy** π from state \mathbf{s} . The **optimal policy** π^* maximizes this value, often computed using **Value Iteration** or **Policy Iteration** algorithms [15]:

$$V^{\pi^*}(s) = \max_a \left[\sum_{s'} P(s'|s, a) R(s, a) + \gamma V^{\pi^*}(s') \right].$$

MDPs have been widely used in robotics for path planning, exploration, and navigation [3]. They enable robots to compute optimal policies for sequential decision problems, making them particularly effective for grid-world environments where the system must balance exploration and exploitation.

However, one major limitation of MDPs is their **computational complexity** in real-time applications, especially in large environments. Since MDPs rely on full knowledge of transition probabilities and rewards, they struggle with dynamic environments where state transitions may change unpredictably. This motivates the need for predictive control to enhance real-time adaptability.

2.2. Predictive Control for Mobile Robots

Predictive control, particularly Model Predictive Control (MPC), has emerged as a powerful approach for real-time motion planning and trajectory tracking in robotics [2]. Unlike MDPs, which focus on long-term reward optimization, MPC formulates an optimal control problem over a finite prediction horizon and continuously updates actions based on real-time sensor data.

MPC solves an optimization problem at each time step to minimize a cost function J while satisfying system constraints:

$$J = \sum_{k=0}^N \left[x_k^T Q x_k + u_k^T R u_k \right]$$

where: J is the cost function,

x_k represents the state vector at time step k ,

u_k represents the control input at time step k ,

Q and R are weight matrices that penalize state deviation and control effort, respectively,

N is the prediction horizon.

Following [14], we adapt the equation for this context.

MPC predicts future states using the system dynamics:

$$x_{k+1} = f(x_k, u_k).$$

Subject to constraints:

$$u_{min} \leq u_k \leq u_{max}, \quad x_{min} \leq x_k \leq x_{max}.$$

Having a predictive capability allows MPC to dynamically adjust robot actions, making it highly effective for applications such as:

- Obstacle avoidance in dynamic environments [12].
- Multi-robot coordination, ensuring collision-free paths [13].
- Real-time trajectory planning in complex terrains [11].

2.3. Combining MDPs and Predictive Control

Although MDPs provide a structured approach for high-level decision-making, they lack adaptability in real time. While MPC excels at short-term control and constraint handling, it lacks an inherent ability to model long-term decision-making.

Integrating MDPs with MPC leverages the advantages of both:

- MDPs generate an optimal policy for global navigation based on reward optimization.
- MPC executes the policy in real time while adapting to dynamic changes.

Hybrid approach allows for robust decision-making and efficient trajectory execution, particularly in dynamic maze exploration and autonomous navigation scenarios. The following sections explore how this integration can enhance mobile robot performance.

3. Case Study: Maze Exploration with MDPs

3.1. System Description

A mobile robot explores a maze autonomously in a grid world environment, where each cell represents a state. Markov Decision Process (MDP)-based algorithms define the transitions between states, guiding the robot's decision-making [4]. The objective is to enable efficient navigation from a starting position to a goal while avoiding obstacles and optimizing movement based on predefined rewards.

- Hardware Setup: The physical robot consists of different components, as shown in Fig. 1, mainly:
 - Microcontroller: The ESP32 microcontroller processes the MDP algorithm and controls the robot's movement.
 - Sensors:
 1. Ultrasonic sensor: Obstacle detection relies on the HC-SR04 sensor.
 2. Camera module: A separate Sony IMX298 camera module connects to the Raspberry Pi using the MIPI CSI-2 interface, then transmits data to the ESP32 microcontroller via Wi-Fi for processing.

Object detection techniques were employed to distinguish the robot from the environment, and a localization module processed this data for accurate mapping [7].

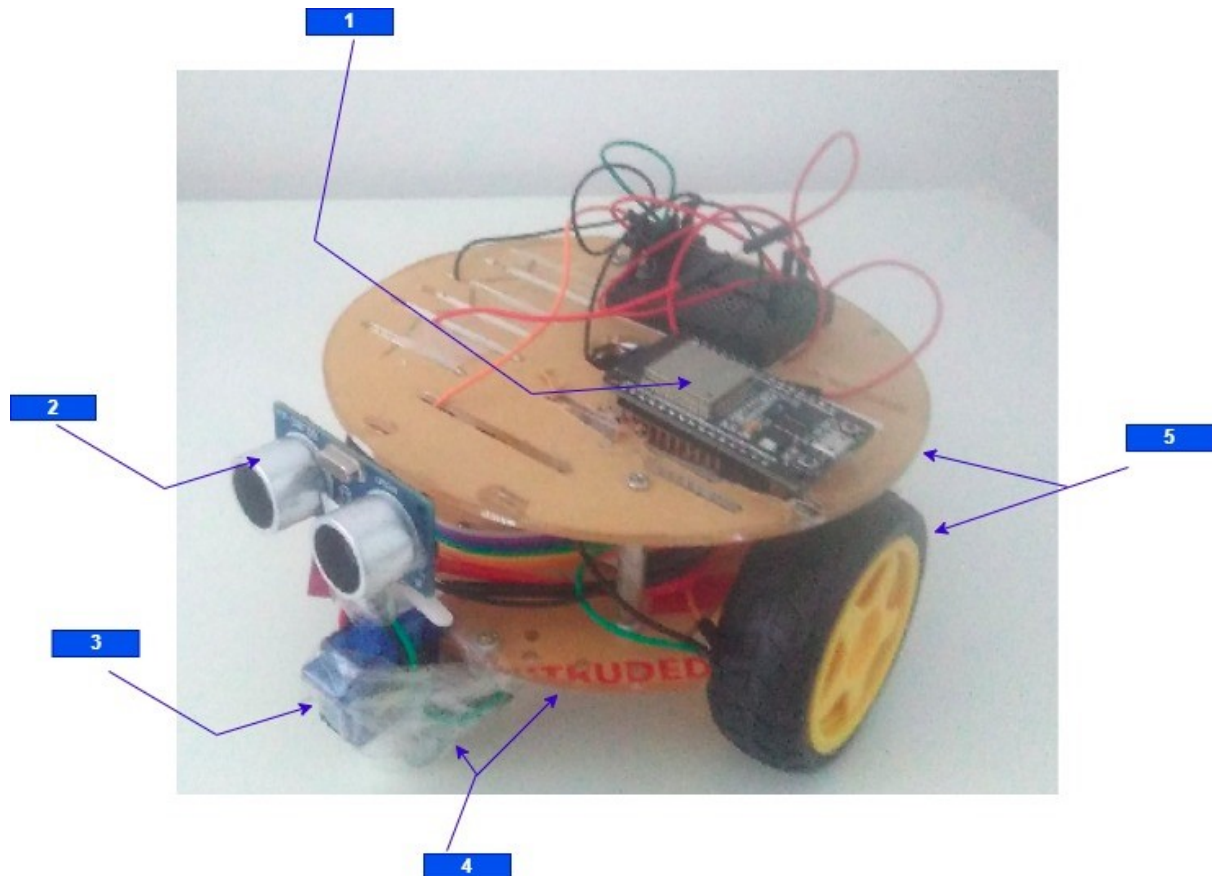


Figure 1: Physical mobile robot used for MDP-based maze exploration. The legend highlights key components: (1) ESP32 μ -controller, (2) Ultrasonic-sensor, (3) Servo-motor, (4) Two castor wheels, and (5) Two primary dual shaft DC motor-driven wheels.

- Motors Driver: Dual-shaft DC motors with a motor driver for precise motion control over movement [1]. Along with two castor wheels, a freely rotating wheel supports the robot's weight and enables smooth, multi-directional movement.
- Software Algorithm Implementation:
 - MDP-Based Decision Making: The robot uses an MDP framework to determine optimal actions in each state.
 - Policy Iteration Value Iteration Algorithms: These methods compute the best navigation policy based on state transitions and rewards.
 - Localization Mapping: A vision-based system helps in state estimation and tracking the robot's movement.
 - Combine a left-hand rule maze exploration algorithm to optimize performance and minimize the robot's rotation time.
- Grid-World Representation:

The environment is modeled as a 3x4 discrete grid-world maze with defined start, goal, and obstacle states, as shown in Fig. 2. The goal was to determine an optimal policy for the robot to navigate from the start state to the goal state while avoiding obstacles and maximizing rewards where:

1. Each cell represents a state (position in the maze). * State transitions are probabilistic, accounting for uncertainties in movement.
2. An agent assigns rewards to different states:
 - +1 for reaching the goal,
 - -1 for entering an obstacle,
 - 0 for intermediate steps

3.2. Experimental Results

We conducted experiments in physical and virtual environments to validate the implementation of MDP-based maze exploration. We tested the robot in a 3x4 grid-world maze and a more extensive virtual 6x8 grid-world environment. The key results are summarized below:

3.2.1. 3x4 Grid-World Environment

In the physical setup, the robot successfully navigated the 3x4 maze, which associates one obstacle (inaccessible) state and two terminal states where the episode ends (reward of +1 or -1) out of the twelve states (cells of the grid) in total, achieving the following outcomes:

- **Convergence of Policy:** Figure 3 shows that the MDP policy iteration algorithm converged after 11 iterations, demonstrating efficient policy computation in small environments [2].
- **Optimal Navigation Path:** The robot followed the computed optimal policy, avoiding obstacles and reaching the goal state. The resulting path minimized cumulative costs and maximized rewards.
- **Localization Performance:** Localization performance was enhanced by the vision-based localization module, which accurately identified the robot's position in most cases and facilitated smooth navigation.

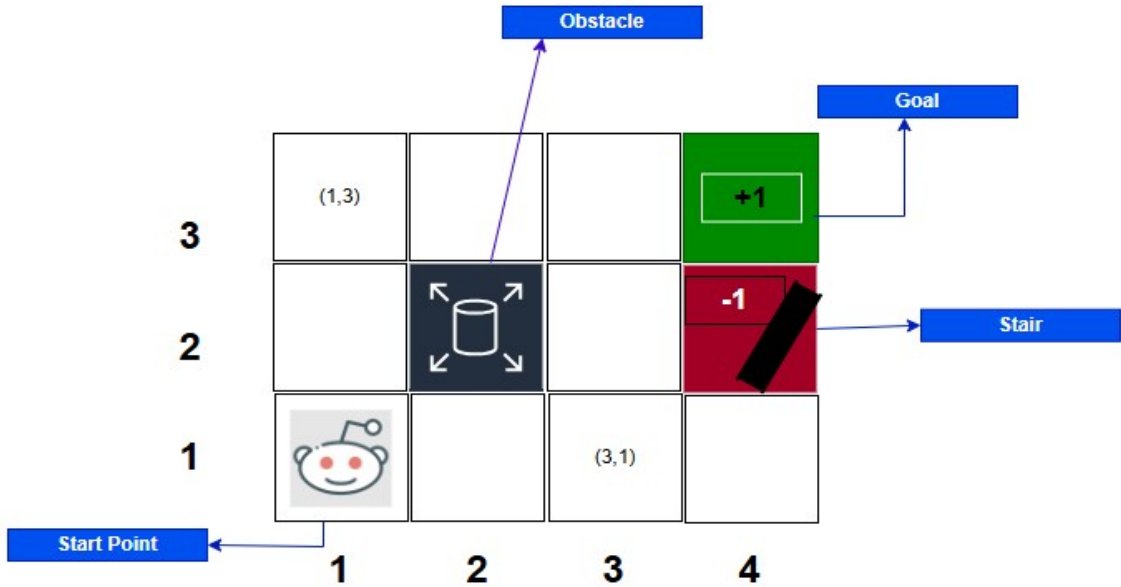


Figure 2: 3x4 Grid-World Environment with different states (Created by the authors based on [15])

3.2.2. 6x8 Grid-World Environment

To evaluate the scalability of the proposed Markov Decision Process (MDP)-based exploration strategy. We tested the system in a more enormous 6x8 virtual maze. The environment consists of 48 states, incorporating:

- There are nine obstacle (inaccessible) states, which refer to areas with obstacles (walls) where the robot cannot traverse.
- There are three terminal states, each with assigned rewards: one positive goal state and two negative penalty states.

Virtually, the robot explored the maze using MDP as its primary algorithm to decide the motion from the current cell to the next potential cell, along with the Left-hand Rule maze exploration algorithm to guide the robot during unwanted maneuvers. The Maze exploration algorithm does not affect either the optimal policy that emerged or the efficiency matrix. Overall, after a short time stamp, the optimal policy generated the as shown in Fig. 4. The final policies for both setups are included to provide a visual understanding which demonstrated the following key observations:

- **Policy Convergence:** The optimal policy was computed after 19 iterations, indicating increased computational demands for larger environments [4]. Since the 6x8 grid-world is 4 times larger than the 3x4 grid-world (48 states vs. 12 states), if the system scaled linearly, we would

expect 44 iterations. However, with the help of the maze exploration algorithm, the system converged into 19 iterations instead of 44. The percentage optimization is 56.82%.

- **Optimal Policy Map:** The computed policy effectively directed the robot to navigate the maze while avoiding prohibited cells. The policy map provided apparent direction vectors for each state. As a result, the optimal policy demonstrates the final, accepted flow that guides the robot reaching the goal state from any permissible cell in the maze.
- **Efficiency Metrics:** Increasing the maze size resulted in a corresponding increase in total computation time, highlighting the necessity for optimization techniques to enhance performance in larger-scale environments. A better strategy emerges from the need to achieve optimal flow convergence in a maze containing various cell types.

We included figures to illustrate the convergence plots, reward values, and final policies for both setups, helping to provide a clear visual understanding of the results.

3.3. Discussion

The results demonstrate the effectiveness of MDP-based methods for maze exploration and navigation. However, several challenges and opportunities for improvement were identified:

3.3.1. Strengths

- **Policy Accuracy:** The MDP algorithms generated reliable policies that guided the robot effectively, even in complex environments.
- **Scalability:** The approach scaled well to larger mazes, demonstrating robustness in generating optimal policies for various grid sizes.
- **Flexibility:** Integrating vision-based localization and sensor data enables the system to successfully facilitate real-world navigation.

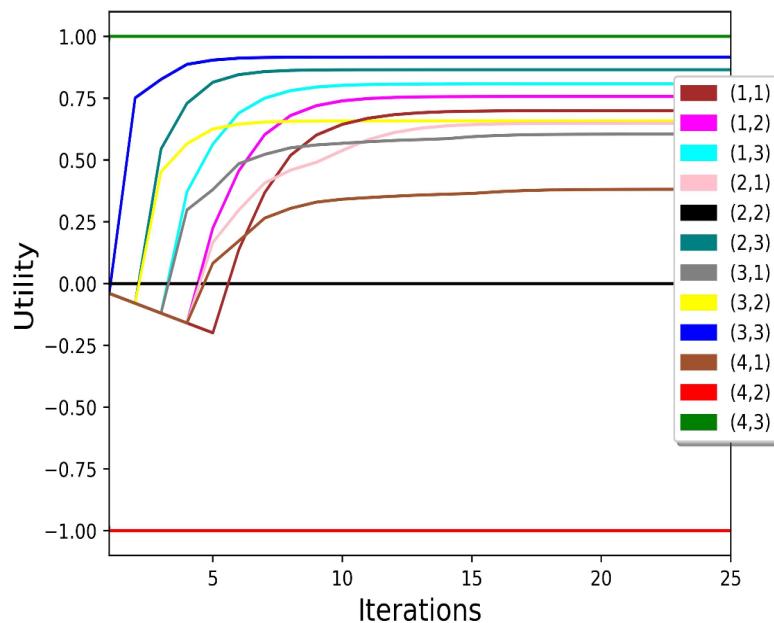


Figure 3: Convergence of the MDP policy iteration algorithm in the 3x4 grid-world environment.

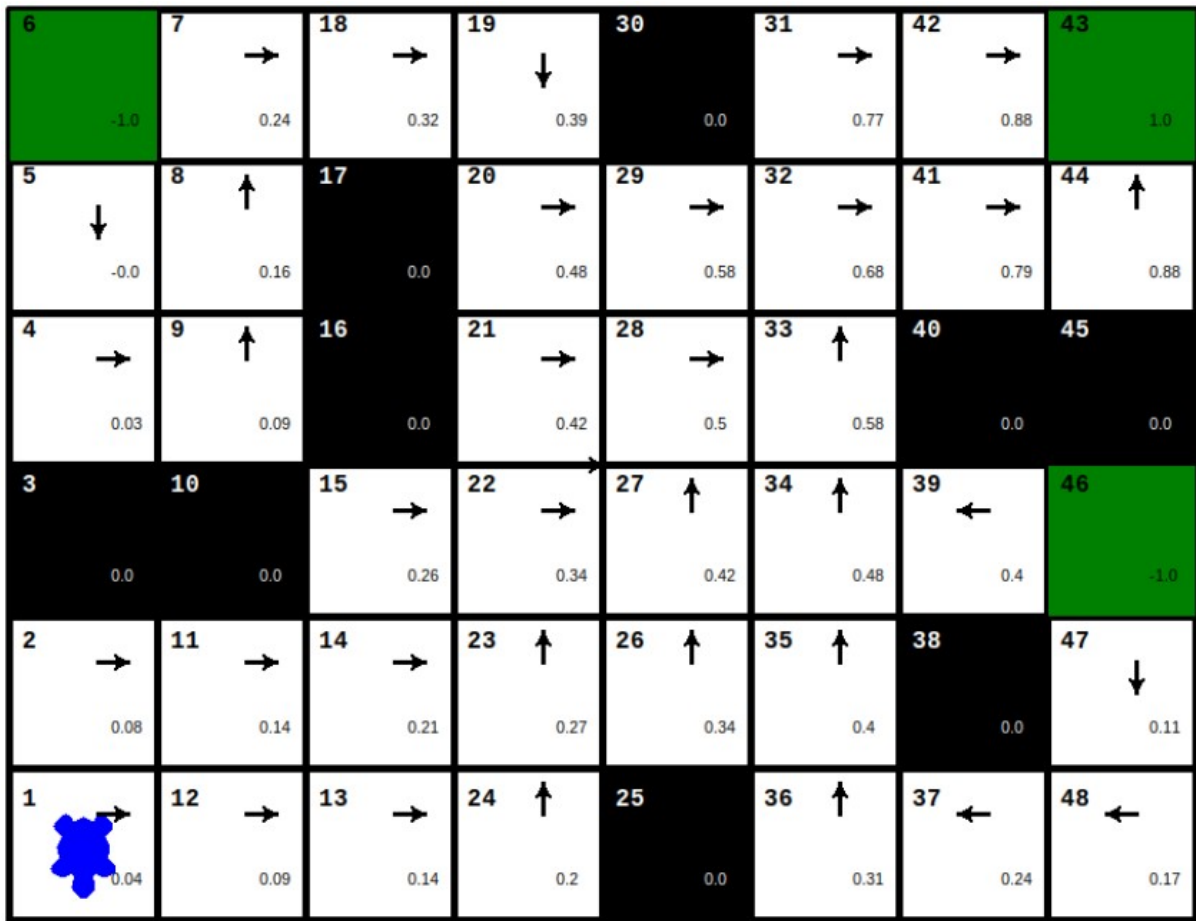


Figure 4: MDP-based Maze Exploration in a 6x8 grid-world. The figure shows the reward and policy map, where each cell represents the state of the environment. The nine black cells indicate obstacles that the robot cannot enter. The green cells represent terminal states, with rewards of -1, 1, and a final state with a +1 reward. The policy map displays optimal action for each state, guiding the robot's exploration in the maze.

4. Connecting MDPs to Predictive Control

4.1. Advantages of Predictive Control for Mobile Robots

Predictive control techniques, such as Model Predictive Control (MPC), have demonstrated significant advantages in addressing real-time adaptability and constraint handling in mobile robotics. Unlike MDPs, which focus on long-term decision-making through reward optimization, MPC excels in short-term trajectory planning by continuously predicting future states and adjusting control inputs accordingly [1,6].

Many regard MPC as one of the most effective methods for controlling autonomous systems under constraints [9]. Its ability to incorporate physical limitations (e.g., motor torque, velocity) and maintain smooth trajectories makes it a valuable complement to MDP-based approaches [2]. Its predictive nature allows the system to compute optimal control actions at each step by solving a constrained optimization problem [10]. It is particularly effective for dynamic environments where robots must respond to changes such as moving obstacles or time-varying conditions [16]. Applications of MPC in mobile robotics include:

- Obstacle avoidance in dynamic environments plays a critical role in real-time navigation [11].
- Real-time trajectory planning for autonomous vehicles is crucial for ensuring safe and efficient navigation [12].
- Coordinated control is essential for multi-robot systems to function optimally [13].

4.2. Challenges in MDP-Based Systems

While MDPs provide an optimal policy for high-level decision-making, they encounter several limitations when applied to real-world robotic systems:

- **Real-Time Constraints:** The iterative computation of policies in MDPs can lead to delays, especially in larger environments, limiting their applicability for fast-changing scenarios [7].
- **Dynamic Environments:** MDPs lack an inherent design for handling dynamic changes, such as moving obstacles or sudden environment updates [2].
- **Trajectory Execution:** Translating discrete state-action policies into smooth, continuous motion trajectories can be challenging without additional control layers [9].

4.3. Proposed Integration of MDPs and Predictive Control

Integrating MDPs with predictive control offers a promising approach to leverage the strengths of both methods [16]. The proposed framework involves:

- **MDP for High-Level Planning:** Use MDPs to generate optimal policies based on long-term goals and rewards. These policies provide a high-level decision-making framework for robots [4].
- **MPC for Low-Level Control:** Employ MPC to execute the MDP-generated policies in real time, ensuring smooth trajectory tracking and adherence to system constraints [8].
- **Feedback Loop:** Integrate a feedback mechanism so MPC informs the MDP of environmental changes, enabling policy adaptation.

4.4. Potential Benefits of Integration

The integration of MDPs and MPC can address the limitations of standalone methods while enhancing overall system performance:

- **Real-Time Adaptability:** MPC's predictive capabilities enable rapid responses to dynamic changes, complementing MDPs' high-level planning [10].
- **Trajectory Optimization:** MPC ensures smooth and efficient trajectory execution, translating discrete MDP policies into actionable continuous motion [9].
- **Scalability and Robustness:** The combined approach allows scalable application to complex environments while maintaining robustness to uncertainties and disturbances [11].

4.5. Applications for Combined Methods

The integration of MDPs and predictive control has broad applications in mobile robotics, including:

- **Autonomous Navigation:** Robots navigating warehouses, hospitals, or urban environments can benefit from the combined framework for efficient and adaptive path planning.
- **Multi-Robot Coordination:** Predictive control can optimize interactions between robots in collaborative tasks, while MDPs ensure high-level task allocation [13].
- **Dynamic Obstacle Avoidance:** The feedback mechanism between MDPs and MPC can handle real-time updates to avoid moving obstacles effectively.

5. Future Work and Conclusion

The findings from this study highlight several key areas for further research and improvement. Future work should focus on addressing the current limitations of MDP-based maze exploration and predictive control integration, including the following aspects:

- **Developing Hybrid MDP-MPC Systems:** While MDPs provide an effective framework for high-level decision-making [1], they lack real-time adaptability. Conversely, Model Predictive Control (MPC) excels in trajectory tracking but does not inherently optimize long-term decision-making [12]. Future work should focus on designing hybrid systems that leverage MDPs for

strategic planning and MPC for real-time control, ensuring a seamless balance between computational efficiency and adaptability in dynamic environments.

- **Enhancing Localization Accuracy Through Sensor Fusion:** One of the primary challenges observed in this study is the reliance on vision-based localization, which is susceptible to errors under poor lighting conditions. Future research should explore multi-sensor fusion techniques, incorporating data from LiDAR, inertial measurement units (IMUs), and ultrasonic sensors to improve localization robustness. Advanced filtering techniques, such as Kalman Filters or Particle Filters, can further enhance state estimation accuracy [11].
- **Optimizing Computational Efficiency for Real-Time Applications:** MDP-based decision-making suffers from scalability issues when applied to large or dynamic environments. Future efforts should explore reinforcement learning approaches, such as Q-learning or Deep Q-Networks (DQNs), to approximate value functions efficiently. Additionally, parallel computing and GPU acceleration could be utilized to speed up policy computation and real-time adaptability [3].
- **Application in Real-World Scenarios:** Future studies should validate the proposed hybrid MDP-MPC system in real-world environments beyond simulated grid-world setups. Potential applications include warehouse automation, autonomous navigation in urban settings, and search and rescue missions, where adaptive decision-making and precise control are crucial [13].
- **Improving Obstacle Avoidance Strategies:** The current MDP framework assumes a static environment. However, real-world navigation often involves dynamic obstacles. Future research should focus on integrating dynamic obstacle avoidance mechanisms using predictive models and real-time environmental perception [11].

6. Conclusion

This study explored the integration of Markov Decision Processes (MDPs) with predictive control techniques for mobile robot trajectory tracking and maze exploration. Through a case study, we demonstrated that while MDPs provide an effective framework for navigation in structured environments [4], they face limitations in real-time adaptability. To address these challenges, we examined how Model Predictive Control (MPC) can enhance trajectory tracking performance by dynamically adjusting control actions in response to environmental changes [12].

Our findings suggest that combining MDPs with predictive control can significantly improve the efficiency and adaptability of autonomous navigation systems. By leveraging the strengths of both methods, robots can achieve optimal decision-making while maintaining real-time responsiveness. The proposed approach has potential applications in autonomous robotics, warehouse automation, and dynamic path planning for mobile robots operating in uncertain environments [13].

Future research should focus on enhancing localization accuracy, optimizing computational efficiency, and applying the hybrid MDP-MPC framework in real world robotic systems. The integration of reinforcement learning techniques [2] and sensor fusion strategies [11] could further improve performance, making mobile robots more capable of handling complex, real-world navigation tasks.

In conclusion, this work revisited MDP-based maze exploration and highlighted its potential when combined with predictive control for mobile robot trajectory tracking.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] M.L.Puterman. "*Markov Decision Processes: Discrete Stochastic Dynamic Programming*.", John Wiley & Sons, 2005. URL: <https://doi.org/10.1002/9780470316887>

- [2] R. S. Sutton and A. G. Barto. “*Reinforcement Learning: An Introduction*”, MIT Press, 2017. URL: <https://doi.org/10.5555/3312046>
- [3] E. Alpaydin. “*Introduction to Machine Learning.*”, MIT Press, 2014. URL: <https://ieeexplore.ieee.org/book/6267367>
- [4] N. Privault. “*Understanding Markov Chains: Examples and Applications.*”, Springer, 2013. URL: <https://doi.org/10.1007/978-981-13-0659-4>
- [5] Q. Hu and W. Yue. “*Markov Decision Processes with Their Applications.*”, Springer, 2008. URL: <http://dx.doi.org/10.1007/978-0-387-36951-8>
- [6] E. A. Feinberg and A. Shwartz. “*Handbook of Markov Decision Processes.*” Springer, 2002. URL: <http://dx.doi.org/10.1007/978-1-4615-0805-2>
- [7] X. Wang, X. Wang, and D. M. Wilkes. “*Machine Learning-Based Natural Scene Recognition for Mobile Robot Localization in an Unknown Environment.*”, Springer, 2019. URL: <http://dx.doi.org/10.1007/978-981-13-9217-7>
- [8] [8] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. “Constrained model predictive control: Stability and optimality.”, *Automatica*, 36(6):789–814, 2000. URL: [https://doi.org/10.1016/S0005-1098\(99\)00214-9](https://doi.org/10.1016/S0005-1098(99)00214-9)
- [9] E. E. F. Camacho and C. Bordons. “*Model Predictive Control.*” Springer, 2013. URL: <https://doi.org/10.1007/978-0-85729-398-5>
- [10] [10] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. “*Model Predictive Control: Theory and Design*”. Nob Hill Publishing, 2009. URL: <https://www.nobhillpublishing.com/mpc/>
- [11] X. Qian, J. R. Akella, and H. A. Ghasemi> “Adaptive Model Predictive Control for Obstacle Avoidance in Dynamic Environments.”, *IEEE Transactions on Robotics*, 2019, vol. 35, no. 2, pp. 431–446. URL: <https://doi.org/10.48550/arXiv.2303.15869>
- [12] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat. “Predictive Active Steering Control for Autonomous Vehicle Systems,” *IEEE Transactions on Control Systems Technology*, 2007, vol. 15, no. 3, pp. 566–580. URL: <https://doi.org/10.1109/TCST.2007.894653>
- [13] M. Turpin, N. Michael, and V. Kumar. “CAPT: Coordinated path planning for multiple robots.”, *The International Journal of Robotics Research*, 2014, 33(9):980–999. URL: <https://doi.org/10.1177/0278364914525241>.
- [14] William C. Cohen. “Optimal control theory—an introduction. Control.”, Prentice-Hall, 1971, vol. 17, pp. 1018. URL: <https://doi.org/10.1002/aic.690170452>.
- [15] J. Russell and P. Norvig. “*Artificial Intelligence: A Modern Approach (International Edition).*”, Pearson, 2021. URL: <https://elibrary.pearson.de/book/99.150005/9781292401171>
- [16] J. Li, J. Sun, L. Liu, and J. Xu. “Model predictive control for the tracking of autonomous mobile robot combined with a local path planning,” *Measurement and Control*, 2021, vol. 54, no. 9-10, pp. 1319–1325. URL: <https://doi.org/10.1177/00202940211043070>